ADAPTIVE NONLINEAR AUTOPILOT FOR ANTI-AIR MISSILES

Michael B. McFarland*
Air Force Research Laboratory
101 W. Eglin Blvd, Ste. 341
Eglin AFB, Florida, USA 32542

Donald T. Stansbery*
QuesTech
214 Government Street
Niceville, Florida, USA 32578

Abstract

The prevalent method of synthesizing nonlinear missile autopilots is by gain-scheduling linear designs. Although this approach has proven successful in numerous applications, the desire to continually improve performance without incurring additional cost suggests the need for a new design paradigm. An opportunity to address this need has been identified from previous research which employed neural network technology to augment approximate dynamic inversion controllers. In the one architecture a neural network adaptively cancels linearization errors through on-line learning, which may be accomplished by a weight update rule derived from Lyapunov theory. This effectively guarantees stability of the closed-loop system. This paper concerns a similar implementation in which neural networks function instead to improve command tracking of gain-scheduled control laws. This theoretical development is then specialized to the problem of synthesizing a bank-to-turn autopilot for an agile anti-air missile. Finally, the resulting hybrid control law is demonstrated in a nonlinear simulation and its performance is evaluated relative to that of the unaugmented gain-scheduled autopilot.

Introduction

Historically, the field of missile autopilot design has been dominated by the technique of gain-scheduling. This approach, documented in Ref. 1, is based on linearizing the vehicle's nonlinear equations of motion about several operating conditions. Linear controllers are then designed for each of these points in the flight envelope, and their gains are scheduled as a function of flight condition to yield a full-envelope nonlinear autopilot. Although this approach has proven successful in numerous practical applications, it does have certain disadvantages. Theoretical investigations of gain-scheduling such as the one in Ref. 2 have shown that controller gains should be scheduled as functions of slowly varying quantities if favorable performance and robustness properties of the individual point designs are to be preserved in the overall scheduled implementation. Control designers are increasingly faced with unprecedented performance requirements which necessitate a large number of point designs to form an adequate gain schedule. A large gain schedule not only requires more time and labor to design, it also creates the need for more wind-tunnel testing and additional computation and storage capacity in the flight control processor.

A potential alternative to the present trend toward more complicated gain-scheduled designs may be found in previous research using neural network technology to augment approximate dynamic inversion controllers. Ref's 3 and 4 are among the examples of this approach to nonlinear control design. The development to be presented in this paper, however, is based on the architecture of Ref. 5, in which a neural network adaptively cancels linearization errors through on-line learning. Said learning is accomplished by a weight update rule derived from Lyapunov theory, guaranteeing stability of the closed-loop system.

This control design methodology was shown to perform exceptionally in nonlinear simulation studies of fixed-wing and rotary-wing aircraft in Ref's. 5 and 6, respectively. In Ref. 7, the same technique was used with minor modifications to design a nonlinear bank-to-turn autopilot for the agile missile described in Ref. 8. Simulation studies of high angle-of-attack maneuvers indicated that the neural network-based adaptive nonlinear controller achieved performance comparable to that of a gain-scheduled autopilot with reduced development effort. This led to further research centered on more complicated multi-layer neural network architectures in Ref. 9 and robustness to linear unmodeled dynamics in Ref. 10. This paper addresses the question of whether the same neural network technology could be used to augment an existing gain-scheduled autopilot rather than an approximate nonlinear design. The preliminary Results presented in Ref. 11 indicated that by starting with a gain-scheduled

*Aerospace Engineer. Member, AIAA.

design and introducing an adaptive neural network one may be able not only to reduce errors associated with operating at conditions between design points, but also enlarge the flight envelope beyond what is currently covered by the gain schedule. While the development in Ref. 11 considered only the simple case of neural networks with no hidden layers, the current investigation employs the more powerful single-hidden-layer architecture of Ref. 9.

This paper begins by presenting a general architecture for integrating a gain-scheduled control law with an adaptive neural network with the objective of improving command following. Next, this methodology is applied in the specific context of a missile autopilot design problem. An existing gain-scheduled autopilot is augmented using neural networks, and nonlinear simulation results are illustrate the effectiveness of the resulting design. Finally, conclusions and directions for future research are presented.

Control Design Methodology

The following development concerns a modification of the technique presented in Ref. 11 for augmenting gain-scheduled controllers using neural networks. The feature which distinguishes this result from the previous efforts is that it is admits the important class of neural networks with one hidden layer composed of sigmoidal neurons.

Problem Formulation

Although the development may readily be generalized to the multi-input, multi-output (MIMO) case, we will restrict our attention to single-input, single-output (SISO) case for simplicity of presentation. Consider a second-order SISO nonlinear system:

$$\ddot{x} = f(x,\dot{x}) + g(x,\dot{x})u \tag{1}$$

where $x(t), \dot{x}(t) \in \Re$ are the state variables, $u(t) \in \Re$ is the control variable, and f and g are sufficiently smooth nonlinear mappings from $\Re \times \Re$ into $\Re$. Further, suppose that the desired response of the closed-loop system is modeled by a linear second-order transfer function $G_m(s)$:

$$G_m(s) = \frac{x_m}{r} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} \tag{2}$$

where ω_n and ζ denote the natural frequency and damping ratio, respectively, of the desired response. Here, x_m represents the state of the desired response model and r denotes an externally applied reference command. Note also that Eq. (2) may be expressed in differential equation form as follows:

$$\ddot{x}_m = -\omega_n^2 x_m - 2\zeta\omega_n \dot{x}_m + \omega_n^2 r \tag{3}$$

Gain-Scheduled Linear Control Design

In an effort to achieve this desired response, a gain-scheduled controller may be synthesized by linearizing Eq. (1) about several operating points and employing any suitable linear control scheme. The details of this design will be omitted here, as we elect instead to focus on the use of neural networks and adaptive control theory to augment an existing controller. In the interest of simplicity, the current development is based on a gain-scheduled linear controller with the simple proportional-plus-derivative (PD) form:

$$u_{GS} = k_p(r - x) + k_d(\dot{r} - \dot{x}) \tag{4}$$

where the gains $k_p > 0$ and $k_d > 0$ are functions of the state x. In practice, these gains may depend additionally on other system parameters, and integral action may also be included.

Assume that applying the control in Eq. (4) to the dynamics in Eq. (1) yields the closed-loop system:

$$\begin{aligned} \ddot{x} &= f(x,\dot{x}) + g(x,\dot{x})u_{GS} \\ &= -\omega_n^2 x - 2\zeta\omega_n \dot{x} + \omega_n^2 x_c + \Delta(x,\dot{x},r,\dot{r}) \\ &= f_m(x,\dot{x},r) + \Delta(x,\dot{x},r,\dot{r}) \end{aligned} \tag{5}$$

where an error function (Δ) has been defined by $\Delta(x,\dot{x},r,\dot{r}) = f(x,\dot{x}) + g(x,\dot{x})u_{GS} - f_m(x,\dot{x},r)$. This error represents the deviation of the closed-loop dynamics from the desired response model. It is important to note that Eq. (5) implicitly assumes that the closed-loop system is in fact second-order. While this condition may be somewhat restrictive, it is unavoidable in the current implementation. Further study is necessary to precisely characterize the implications of this assumption.

Neural Network-Based Adaptive Control

We are now prepared to augment the system in an effort to eliminate the effect of the error function $\Delta(x,\dot{x},r)$. To accomplish this, we inject an additional control signal ($-u_{ad}$) at the plant input. This results in the modified system described by

$$\ddot{x} = f_m(x,\dot{x},r) + \Delta(x,\dot{x},r,\dot{r}) - g(x,\dot{x})u_{ad} \tag{6}$$

Figure 1 depicts the augmented system of Eq. (6).

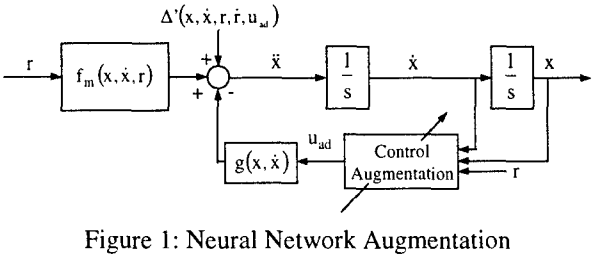

Figure 1: Neural Network Augmentation

Defining the error variable $\tilde{x} = x_m - x$, we may subtract Eq. (6) from Eq. (3) to obtain the following second-order equation for the error dynamics:

$$\ddot{\tilde{x}} = -\omega_n^2 \tilde{x} - 2\zeta\omega_n \dot{\tilde{x}} + g(x,\dot{x})u_{ad} - \Delta(x,\dot{x},r,\dot{r}) \qquad (7)$$

Further, define a pseudo-control v using the following control transformation:

$$v = \hat{g}(x,\dot{x})u_{ad} \quad \Leftrightarrow \quad u_{ad} = \hat{g}^{-1}(x,\dot{x})v \qquad (8)$$

where $\hat{g}(x,\dot{x})$ is a nonsingular approximation of the unknown control influence matrix $g(x,\dot{x})$. We may then rewrite Eq. (7) and substitute to obtain

$$\begin{aligned}\ddot{\tilde{x}} &= -\omega_n^2 \tilde{x} - 2\zeta\omega_n \dot{\tilde{x}} + \hat{g}(x,\dot{x})u_{ad} + [g(x,\dot{x}) - \hat{g}(x,\dot{x})]u_{ad} \\ &\quad - \Delta(x,\dot{x},r,\dot{r}) \\ &= -\omega_n^2 \tilde{x} - 2\zeta\omega_n \dot{\tilde{x}} + v \\ &\quad - \{\Delta(x,\dot{x},r,\dot{r}) - [g(x,\dot{x}) - \hat{g}(x,\dot{x})]\hat{g}^{-1}(x,\dot{x})v\} \\ &= -\omega_n^2 \tilde{x} - 2\zeta\omega_n \dot{\tilde{x}} + v - \Delta'(x,\dot{x},r,\dot{r},v)\end{aligned} \qquad (9)$$

where all uncertain nonlinear terms have been collected into one uncertainty, denoted by Δ'. Note that if $\hat{g}(x,\dot{x}) = g(x,\dot{x})$ and in particular if $\hat{g}(x,\dot{x}) = g(x,\dot{x}) = 1$, then $\Delta' \equiv \Delta$. It will later be useful to define the pseudo-control as the sum of two contributions as follows:

$$v = v_{ad} - \bar{v} \qquad (10)$$

where v_{ad} is an adaptive term to be computed using a neural network, and $\bar{v}$ will be designed to account for nonlinear parameterization effects.

A minimal state-space realization of the augmented error system in Eq. (9) is then given by

$$\dot{z} = Az + b[v_{ad} - \bar{v} - \Delta'(x,\dot{x},r,\dot{r},v)] \qquad (11)$$

where z denotes the state vector, A is Hurwitz, and $b^T = [0 \quad 1]$. This is similar to the form of the dynamics considered in Ref. 9. Note that, in contrast with the original gain-scheduled controller, it will be necessary to integrate the state x_m of the desired model online.

We now describe briefly the neural network architecture used to parameterize Δ' and compute v_{ad}. The neural network topology, depicted in Figure 2, is representative of a common class of feedforward networks known as multilayer perceptrons. Because of their universal approximation capability, we will restrict attention to networks with a single hidden-layer of sigmoidal neurons (defined below). In Figure 2, $\bar{x} \in \Re^{N_1}$ denotes the neural network input vector, $V \in \Re^{N_2 \times N_1}$ denotes the matrix of interconnection weights between the input layer and the hidden layer, N_2 is the number of hidden-layer neurons, $w \in \Re^{N_2}$ is the vector of interconnection weights between the hidden and output layers, and y represents the neural network

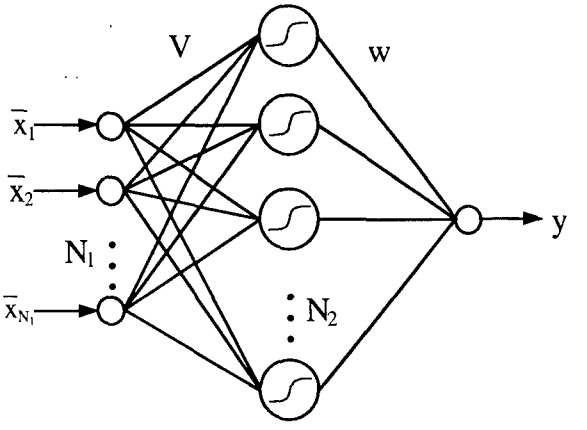

Figure 2: Neural Network with One Hidden Layer

output. The output of the particular multilayer network shown in Figure 2 is written mathematically as

$$y = \sum_{i=1}^{N_2}\left[w_i \sigma\left(\sum_{j=1}^{N_1} v_{ij}\bar{x}_j + \theta_{v_i} \right) + \theta_w \right] \qquad (12)$$

where σ is the hidden-layer activation function, v_{ij} are the first-to-second layer interconnection weights, and w_i are the second-to-third layer interconnection weights. The bias terms θ_w and θ_{vi} represent thresholds. This architecture has N_1 inputs and N_2 hidden-layer neurons. The form of the hidden-layer activation function is a design parameter, but we will consider only the sigmoidal functions with activation potential a:

$$\sigma(z) = \frac{1}{1+e^{-az}} \qquad (13)$$

We may express Eq. (12) in matrix form as

$$y = w^T \sigma(V^T \bar{x}) \qquad (14)$$

where the thresholds are incorporated into the weight matrices as follows:

$$w^T = [\theta_w | w_i] \qquad V^T = [\theta_{v_i} | v_{ij}]$$

Here, we have defined

$$\begin{aligned} y^T &= [y_1 \quad y_2 \quad \cdots \quad y_{N_3}] \\ \bar{x}^T &= [1 \quad \bar{x}_1 \quad \bar{x}_2 \quad \cdots \quad \bar{x}_{N_1}] \\ \sigma(z) &= [1 \quad z_1 \quad z_2 \quad \cdots] \end{aligned}$$

A neural network of the type described above is capable of approximating any smooth function to any desired accuracy, provided the number of hidden-layer neurons N_2 is sufficiently large. This implies that for continuous $\Delta'(x,\dot{x},r,\dot{r},v)$ and any $\varepsilon_N > 0$ there exists a finite N_2 and "ideal" weight matrices W and V such that

$$\Delta' = w^T \sigma(V^T \bar{x}) + \varepsilon(\bar{x}) \qquad (15)$$

with $\|\varepsilon\| \le \varepsilon_N$. Given the parameterization of Δ' described by Eq. (15), an adaptive neural network of the same structure is therefore used to compute v_{ad}.

(8)

$$v_{ad} = \hat{w}^T \sigma\left(\hat{V}^T \bar{x}\right) \tag{16}$$

In this expression, the network weights $\hat{w}(t)$ and $\hat{V}(t)$ are adjustable parameters which will be updated such that v_{ad} approximately cancels the effect of Δ.

An adaptive scheme will be employed to compute the weight matrices $\hat{w}$ and $\hat{V}$ on-line so that the network requires no off-line learning phase. The Lyapunov arguments used to derive stable learning rules also yield the required expression for the pseudo-control term $\bar{v}$. First, assume that the ideal weight matrices are bounded in the sense that $\|w\| \le \bar{w}$ and $\|V\|_F \le \bar{V}$, where $\|\cdot\|_F$ denotes the Frobenius norm, $\|A\|_F^2 = \mathrm{tr}\{A^T A\}$. When not specified otherwise, $\|\cdot\|$ indicates the Euclidean norm. Introducing more compact notation, we may also write

$$Z = [w \quad V] \text{ with } \|Z\|_F \le \bar{Z} \tag{17}$$

The reference command and its first time-derivative are also assumed to be norm-bounded.

Observing the functional dependencies in Eq.'s (11) above and (26) to be presented later, we choose the network input

$$\bar{x}^T = \begin{bmatrix} 1 & z^T & r & \dot{r} & v_{ad} & \|\hat{Z}\|_F \end{bmatrix} \tag{18}$$

which is bounded as follows:

$$\|\bar{x}\| \le c_1 + c_2\|z\| + c_3\|\tilde{Z}\|_F ; c_i > 0 \tag{19}$$

A key result which facilitates the extension of theory previously developed for simple linear-in-parameters neural networks to more complicated neural networks with one hidden layer has been described in Ref. 8. It involves the use of a Taylor series expansion of the hidden-layer output. First, define the error variables $\tilde{w} = \hat{w} - w$, $\tilde{V} = \hat{V} - V$, and $\tilde{Z} = \hat{Z} - Z$. Also, define the hidden-layer output error as follows:

$$\tilde{\sigma} = \hat{\sigma} - \sigma = \sigma\left(\hat{V}^T \bar{x}\right) - \sigma\left(V^T \bar{x}\right) \tag{20}$$

The Taylor series expansion of σ about $\hat{\sigma}$ may then be written as

$$\sigma\left(V^T \bar{x}\right) = \sigma\left(\hat{V}^T \bar{x}\right) - \sigma'\left(\hat{V}^T \bar{x}\right)\tilde{V}^T \bar{x} + O\left(\tilde{V}^T \bar{x}\right)^2 \tag{21}$$

where $\sigma'(\hat{z}) = d\sigma(z)/dz\big|_{z=\hat{z}}$ and $O(\cdot)^2$ denotes those terms with order greater than one. These higher-order terms satisfy Eq. (22):

$$\begin{aligned} O\left(\tilde{V}^T \bar{x}\right)^2 &= \left[\sigma\left(V^T \bar{x}\right) - \sigma\left(\hat{V}^T \bar{x}\right)\right] + \sigma'\left(\hat{V}^T \bar{x}\right)\tilde{V}^T \bar{x} \\ \Rightarrow \left\|O\left(\tilde{V}^T \bar{x}\right)^2\right\| &\le c_4 + c_5\|\tilde{V}\|_F + c_6\|\tilde{V}\|_F\|z\| + c_7\|\tilde{V}\|_F\|\tilde{w}\| \\ & \qquad c_i > 0 \end{aligned} \tag{22}$$

where the inequality follows from the definition of the sigmoid function and properties of vector norms.

In preparation for the Lyapunov analysis, rewrite the error dynamics as follows:

$$\begin{aligned} \dot{z} &= Az + \hat{w}^T\hat{\sigma} - w^T\sigma - \varepsilon - \bar{v} \\ &= Az + \tilde{w}^T\hat{\sigma} - \tilde{w}^T\tilde{\sigma} + \hat{w}^T\tilde{\sigma} - \varepsilon - \bar{v} \end{aligned} \tag{23}$$

Recalling Eq. (20), we may also write

$$\dot{z} = Az + \tilde{w}^T\left(\hat{\sigma} - \hat{\sigma}'\hat{V}\bar{x}\right) + \hat{w}^T\hat{\sigma}'\tilde{V}^T\bar{x} + (d - \bar{v}) \tag{24}$$

where $\hat{\sigma}' = \sigma'\left(\hat{V}^T\bar{x}\right)$ and any extraneous terms have been collected into the disturbance signal, d.

$$\begin{aligned} d &= \tilde{w}^T\hat{\sigma}'V^T\bar{x} + w^T O\left(\tilde{V}^T\bar{x}\right)^2 - \varepsilon \\ \Rightarrow \|d\| &\le C_0 + C_1\|\tilde{Z}\|_F + C_2\|\tilde{Z}\|_F\|z\| + C_3\|\tilde{Z}\|_F^2 \\ & \qquad C_i > 0 \end{aligned} \tag{25}$$

The norm-bound on d follows from Eq.'s (19) and (22) along with the boundedness of ε and $\hat{\sigma}'$ by certain norm inequalities.

For the stability analysis, consider a typical choice of Lyapunov function candidate.

$$V_2 = \frac{1}{2}z^T z + \frac{1}{2\gamma}\tilde{w}^T\tilde{w} + \frac{1}{2}\mathrm{tr}\left(\tilde{V}^T\Gamma_w^{-1}\tilde{V}\right) \tag{26}$$

The Lyapunov analysis is similar to that of Ref. 9, with modifications to account for the structure of the error dynamics in Eq (11). Based on this development, learning rules for the weight matrices are chosen as follows:

$$\begin{aligned} \dot{\hat{V}} &= \dot{\tilde{V}} = -\Gamma_v\left[\bar{x}\left(z^T\hat{w}^T\hat{\sigma}'\right) + \lambda\|z\|\hat{V}\right] \\ \dot{\hat{w}} &= \dot{\tilde{w}} = -\gamma\left[\left(\hat{\sigma} - \hat{\sigma}'\hat{V}^T\bar{x}\right)z^T + \lambda\|z\|\hat{w}\right] \end{aligned} \tag{27}$$

Note that the λ terms in Eq. (27) correspond to the e-modification found in adaptive control literature. These terms provide additional damping which helps to contain the growth of the parameter estimates. In addition, the pseudo-control term $\bar{v}$ in Eq. (10) is defined as

$$\bar{v} = K_Z\left(\|\hat{Z}\|_F + \bar{Z}\right)b^T z \tag{28}$$

where the gain K_Z must be chosen such that

$$K_Z > C_2 \tag{29}$$

in order to effectively dominate nonlinear phenomena associated with parameter errors.

We then conclude that the trajectories of the closed-loop error system are uniformly ultimately bounded provided that, in addition to $\lambda > C_3$, either of the following two inequalities hold:

$$\begin{aligned} \|z\| &> \frac{(\lambda - C_3)C_4^2/4 + C_0}{\underline{\sigma}(A)} = b_z \\ \|\tilde{Z}\|_F &> C_4/2 + \sqrt{C_4^2/4 + C_0/(\lambda - C_3)} = b_Z \end{aligned} \tag{30}$$

where $C_4 = \left(\lambda\bar{Z} + C_1\right)/(\lambda - C_3)$ and $\underline{\sigma}(A)$ is the minimum singular value of A.

Missile Autopilot Application

In this section, the neural network-based adaptive control methodology described above is used to augment an existing gain-scheduled linear autopilot for the agile anti-air missile described in Ref. 8. Studies of high angle-of-attack maneuvering have determined that an appropriate strategy for this vehicle is to design a bank-to-turn autopilot which tracks external guidance commands in angle-of-attack while maintaining near-zero sideslip. There are several possible choices for the roll-channel command, including aerodynamic bank angle (μ), body roll angle (ϕ), body roll rate (p), and stability-axis roll rate ($p_s = p\cos\alpha + r\sin\alpha$). Whenever necessary, the following discussion assumes that the derivatives of external commands are available as outputs of a command rate limiter or pre-filter.

Nonlinear Dynamics & Gain-Scheduled Design

The motion of a symmetric missile about its velocity vector may be described by the following equations.

$$\begin{aligned}\dot{V} &= (X+g_x+T_x)\cos\alpha\cos\beta+(Y+g_y+T_y)\sin\beta \\ &\quad +(Z+g_z+T_z)\sin\alpha\cos\beta \\ \dot{\alpha} &= \frac{1}{V\cos\beta}\left[(Z+g_z+T_z)\cos\alpha-(X+g_x+T_x)\sin\alpha\right] \\ &\quad +q-(r\sin\alpha+p\cos\alpha)\tan\beta \\ \dot{\beta} &= -\frac{1}{V}\left[(X+g_x+T_x)\cos\alpha\sin\beta+(Y+g_y+T_y)\cos\beta\right. \\ &\quad \left.+(Z+g_z+T_z)\sin\alpha\sin\beta\right]+p\sin\alpha-r\cos\alpha\end{aligned} \tag{31}$$

where V, α, and β denote airspeed, angle-of-attack, and sideslip angle while p, q, and r are the body-axis angular rates. The quantities X, Y and Z are the body-axis components of the acceleration due to aerodynamic forces. Similarly, $g_{x,y,z}$ and $T_{x,y,z}$ denote body-axis components of acceleration associated with gravitational effects and thrust, respectively. The moment equations have the form

$$\begin{aligned}\dot{p} &= \frac{L}{I_{xx}} \\ \dot{q} &= \frac{M}{I_{yy}}+\left(1-\frac{I_{xx}}{I_{yy}}\right)pr \\ \dot{r} &= \frac{N}{I_{yy}}-\left(1-\frac{I_{xx}}{I_{yy}}\right)pq\end{aligned} \tag{32}$$

where L, M, and N represent aerodynamic moments about the body axes. Finally, I_{xx} and I_{yy} are rolling and pitching moments of inertia, respectively, and are assumed for simplicity to have constant or slowly varying values.

We will now derive simplified models of Eq's. (31) and (32) which will be used for purposes of linear autopilot design. First we neglect gravitational acceleration, since it is divided by V. Zeroing the lateral dynamics and linearizing the longitudinal dynamics about a particular angle-of-attack α_0 results in

$$\begin{aligned}\dot{\alpha} &= \frac{1}{V}\left(q+Z_\alpha\alpha+Z_{\delta e}\delta_e-Tx\sin\alpha_0+Tz\cos\alpha_0\right) \\ \dot{q} &= \frac{1}{I_{yy}}\left(M_\alpha\alpha+M_q q+M_{\delta e}\delta_e+M_T\right)\end{aligned} \tag{33}$$

where δ_e denotes the pitch control input, M_T is the pitching moment caused by thrust and

$$\begin{aligned}Z_\alpha &= \frac{\partial\dot{\alpha}}{\partial\alpha}=\left[(\partial Z/\partial\alpha-g_x-T_x-X)\cos\alpha\right. \\ &\quad \left.-(\partial X/\partial\alpha-g_z-T_z-Z)\sin\alpha\right]_{\alpha=\alpha_0} \\ Z_{\delta e} &= \frac{\partial\dot{\alpha}}{\partial\delta_e}=\left[(\partial Z/\partial\delta_e)\cos\alpha-(\partial X/\partial\delta_e)\sin\alpha\right]_{\alpha=\alpha_0}\end{aligned}$$

$$M_\alpha=\left.\frac{\partial M}{\partial\alpha}\right|_{\alpha=\alpha_0} \qquad M_q=\left.\frac{\partial M}{\partial q}\right|_{\alpha=\alpha_0} \qquad M_{\delta_e}=\left.\frac{\partial M}{\partial\delta_e}\right|_{\alpha=\alpha_0}$$

Neglecting gravity and the constant term $T_x\sin\alpha_0$, we may write the linear pitch-plane model

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix}=\begin{bmatrix}Z_\alpha/V & 1\\ M_\alpha & M_q\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix}+\begin{bmatrix}Z_{\delta e}/V\\ M_{\delta e}\end{bmatrix}\delta_e \tag{34}$$

A similar process results in the following analogous linear design model for the yaw-roll channel:

$$\begin{bmatrix}\dot{\beta}\\ \dot{p}\\ \dot{r}\end{bmatrix}=\begin{bmatrix}(Y_\beta/V+\sin\alpha_0) & (Y_p/V+\sin\alpha_0) & (Y_r/V+\sin\alpha_0)\\ L_\beta/I_{xx} & L_p/I_{xx} & L_r/I_{xx}\\ N_\beta/I_{yy} & N_p/I_{yy} & N_r/I_{yy}\end{bmatrix} + \begin{bmatrix}Y_{\delta a}/V & Y_{\delta r}/V\\ L_{\delta a}/I_{xx} & L_{\delta r}/I_{xx}\\ N_{\delta a}/I_{yy} & N_{\delta r}/I_{yy}\end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\end{bmatrix} \tag{35}$$

In the example which follows, we will use adaptive neural networks to augment a gain-scheduled autopilot designed for the agile anti-air missile featured in Ref. 8 using linear quadratic regulator theory as described in Ref. 12. The design consists of decoupled pitch and yaw-roll channels, and includes integral control as well as dependence on vehicle mass and Mach number, necessitating minor deviations from the technique described above. The general design philosophy, however, remains unchanged. For simplicity of presentation, we will focus on the longitudinal autopilot, which has the form:

$$\begin{aligned}\delta_e &= k_p(\alpha,M,m)(\alpha_c-\alpha)+k_i(\alpha,M,m)\int_0^t(\alpha_c-\alpha)d\tau \\ &\quad +k_d(\alpha,M,m)q\end{aligned} \tag{36}$$

Note that pitch rate q is used as a state variable rather than $\dot{\alpha}$, since the equation of motion governing

angle-of-attack is not easily expressed in second-order form. By virtue of the relationship between $\dot{\alpha}$ and q, however, the gain-scheduled control defined by Eq. (29) may be seen to bring the closed-loop system into a form similar to Eq. (5).

$$\begin{aligned}\ddot{\alpha} &= -\omega_n^2\alpha - 2\zeta\omega_n\dot{\alpha} + \omega_n^2\alpha_c + \Delta(\alpha,q,\alpha_c) \\ &= f_m(\alpha,q,\alpha_c) + \Delta(\alpha,q,\alpha_c)\end{aligned} \tag{37}$$

Assuming that the roll channel can be regulated effectively by a simple proportional feedback control, an implementation similar to the longitudinal design may be used for the yaw channel.

Neural Network Augmentation

We now introduce scalar neural network augmentation in yaw, pitch, and roll. A feedforward neural network with a single hidden layer is capable of approximately reconstructing the error function (Δ) in each channel and may be used to compute $u_{ad}(t)$ as described previously.

$$\begin{aligned}u_{ad} &= \hat{g}^{-1}v \\ v &= v_{ad} - \bar{v} \\ v_{ad} &= \hat{w}^T\sigma(V^T\bar{x})\end{aligned} \tag{38}$$

Moreover, a stable learning rule is again given by Eq. (27). Various neural network topologies and choices of inputs were considered in this study. The results presented correspond to a neural network with only five sigmoidal hidden-layer neurons. Each neuron has an internal activation potential of a = 0.01. Network inputs were chosen to include z, r, v_{ad}, $\|Z\|_F$, and Mach number along with the customary bias term. Although it does not appear in Eq. (16), Mach number is included here because its variation affects the nonlinear inversion error. This does not alter the proof of stability, since for our purposes Mach number may be considered a bounded, slowly-varying parameter. The network input $\dot{r}$ was omitted in order to reduce the size of the network. The learning rates were chosen to be $\gamma = 500$ and $\Gamma_V = 500\times I$. The damping coefficient in the weight update was assigned the value $\lambda = 0.01$, and the term $\bar{v}$ was neglected for simplicity.

Simulation Results

Based on the results obtained in Ref's 7 and 9, a properly designed neural network should achieve desired performance in the face of uncertainty. These previous efforts describe implementations in which neural networks were employed in conjunction with crudely designed nonlinear control laws derived from approximate dynamic inversion. Since these efforts both perform at least as well as the gain-scheduled autopilot, the combination of neural network and gain-scheduled controller is expected to achieve similar results.

Numerical simulation studies have been performed using a nonlinear, six-degree-of-freedom (6-DOF) simulation of the agile anti-air missile featured in Ref. 8. Figure 3 shows step responses with and without neural network augmentation for an angle of attack command with magnitude 90 degrees. This comparison indicates that the addition of the neural network causes the response to follow the prescribed first-order model.

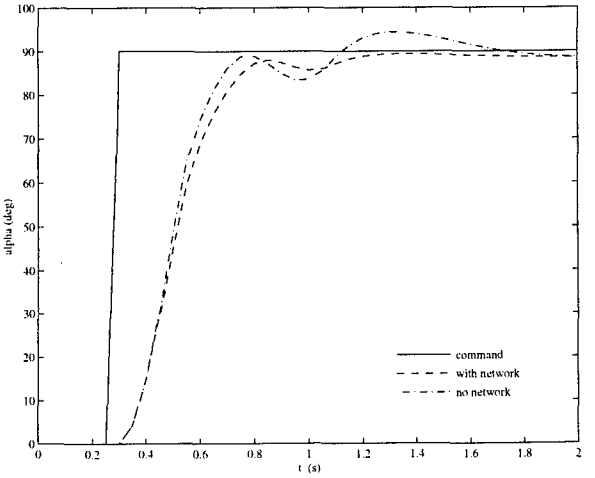

Figure 3: Angle of Attack Response Comparison

A typical engagement geometry for an agile missile of this type is the "head-on merge" scenario illustrated in Figure 4. This engagement geometry represents a planar intercept with both missile and target trajectories evolving primarily in the local horizontal plane. Although individual scalar neural networks were implemented, coupling effects were addressed by including cross-channel variables as network inputs.

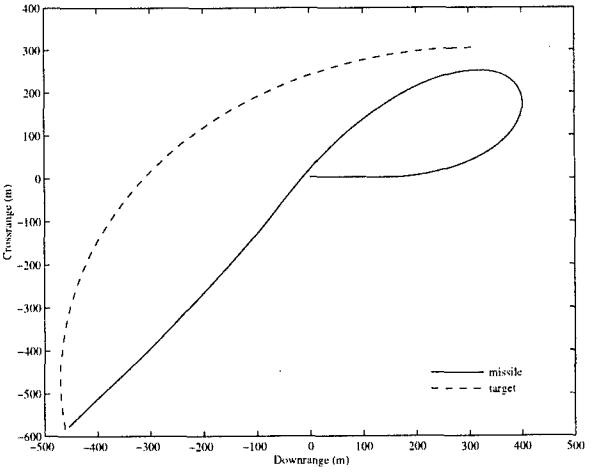

Figure 4: Merge Intercept Trajectory

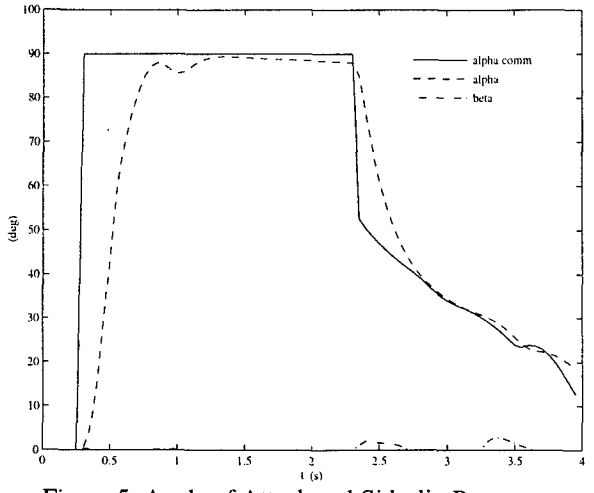

Figure 5: Angle of Attack and Sideslip Response

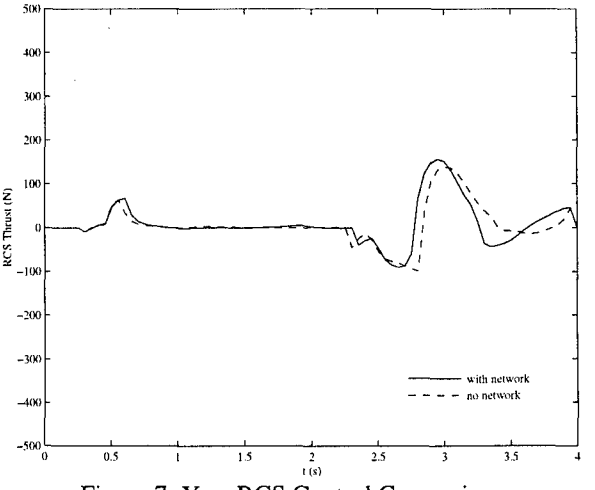

Figure 7: Yaw RCS Control Comparison

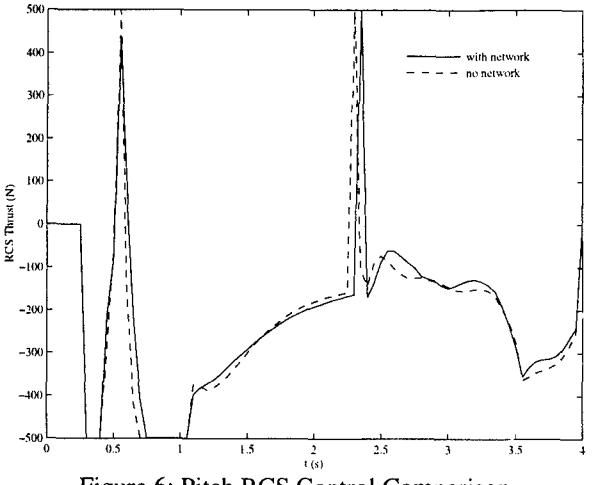

Figure 6: Pitch RCS Control Comparison

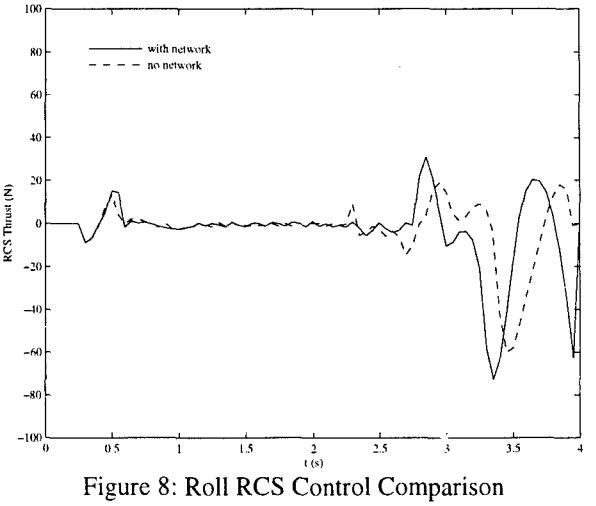

Figure 8: Roll RCS Control Comparison

Figure 5 depicts the angle of attack and sideslip tracking response for the gain-scheduled autopilot with adaptive neural augmentation. The sideslip angle remains within acceptable values, but the overall response of the system is somewhat sluggish. This is an inherent disadvantage of adaptive schemes, adaptation occurs over a small but finite interval. These results are quite favorable, however, especially considering the fact that the design technique outlined above does not explicitly address performance. More research is needed to fully characterize the interaction between the nominal closed-loop dynamics associated with the gain-scheduled control and the dynamics of the adaptive neural augmentation system. As expected, Figures 6-8 indicate that the results achieved necessitated only slight modification of the RCS thruster commands. This is a feature of the neural network implementation, which does not encourage excessive control activity or actuator saturation.

Conclusions

A control design methodology enabling the integration of recently developed neural network-based adaptive control technology with existing gain-scheduled linear controllers has been developed. In the proposed architecture, a neural network adaptively augments plant input signals in order to enforce tracking of a reference model. This development was motivated by previous research involving nonlinear dynamic inversion controllers, which suggested that this design technique could potentially improve both the performance and robustness of existing gain-scheduled control laws. As an example, the problem of designing a longitudinal missile autopilot to track commands in angle-of-attack was considered. Nonlinear 6-DOF simulation results indicated that a neural network added to the longitudinal channel of the gain-scheduled autopilot resulted in improved tracking of a reference model. This effect was achieved by explicitly

accounting for nonlinear dynamics through an adaptive neural network parameterization.

Efforts to implement a similar augmentation system for the roll-yaw autopilot are in progress. In addition, further research is necessary in order to extend this control design methodology to the more general case where the order of the closed-loop system under the action of the original gain-scheduled controller is uncertain. This may necessitate the use of dynamic or recurrent neural network architectures, which include time-delays and are capable of learning to approximate the input-output relationships of dynamical systems. These recurrent architectures are currently the subject of ongoing research in which they are being employed as parameterizations of uncertain dynamical systems for the purpose of adaptive control.

References

1. Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, John Wiley & Sons, Inc., New York, 1991.

2. Shamma, J. S., and Athans, M., "Analysis of Gain Scheduled Control for Nonlinear Plants," *IEEE Transactions on Automatic Control*, Vol. 35, No. 8, pp. 898-907, 1990.

3. Tan, W. K., and Lee, T. H., "A Neural Net Control System With Parallel Adaptive Enhancements," *Proceedings of the American Control Conference*, pp. 61-62, 1992.

4. McDowell, D. M., Irwin, G. W., and McConnell, G., "Online Neural Control Applied to a Bank-to-Turn Missile Autopilot," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1286-1294, 1995.

5. Kim, B. S., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, pp. 26-33, 1997.

6. Leitner, J., Calise, A., and Prasad, J. V. R., "Analysis of Adaptive Neural Networks for Helicopter Flight Controls," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 972-979.

7. McFarland, M. B., and Calise, A. J., "Neural-Adaptive Nonlinear Autopilot Design for and Agile Anti-Air Missile," *AIAA Guidance, Navigation, and Control Conference*, AIAA-96-3914, 1996.

8. Wise, K. A., and Broy, D. J., "Agile Missile Dynamics and Control," *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, AIAA-96-3912, 1996.

9. McFarland, M. B., and Calise, A. J., "Multilayer Neural Networks and Adaptive Nonlinear Control of Agile Anti-Air Missiles," *AIAA Guidance, Navigation, and Control Conference*, pp. 401-410, 1997.

10. McFarland, M. B., and Calise, A. J., "Robust Adaptive Control of Uncertain Nonlinear Systems Using Neural Networks," *Proceedings of the American Control Conference*, pp. 1996-2000, 1997.

11. McFarland, M. B., "Augmentation of Gain-Scheduled Missile Autopilots Using Adaptive Neural Networks," *AIAA Guidance, Navigation, and Control Conference*, Boston, MA, AIAA-98-4491, 1998.

12. Wise, K. A., "Bank-to-Turn Missile Autopilot Design Using Loop Transfer Recovery," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, pp. 145-152, 1990.